

CLAIMS

We claim:

1. A system for generating an optimized binary representation of an object tree, comprising:

an input interface to receive markup language information corresponding to an object tree;

a mapping engine, the mapping engine communicating with the input interface to receive the markup language information and generate an optimized binary representation of the markup language information; and

an output interface, communicating with the mapping engine, the output interface exposing the optimized binary representation to external resources.

2. A system according to claim 1, wherein the markup language information comprises extensible application markup language information.

3. A system according to claim 1, wherein the object tree corresponds to user interface elements.

4. A system according to claim 1, wherein the optimization comprises optimizing the binary representation at least by encoding dimension information in the binary representation.

5. A system according to claim 4, wherein the dimension information comprises at least one of a length field and a width field.

6. A system according to claim 1, wherein the output interface comprises a serial interface.

7. A system according to claim 6, wherein the serial interface communicates a serialized binary representation to a client machine.
8. A system according to claim 1, wherein the external resources to which the binary representation is exposed comprise application programming interfaces.
9. A system according to claim 1, wherein the optimization comprises indexing the first instance of a novel object type in the binary representation.
10. A system according to claim 1, wherein the optimization comprises embedding an identifier to invoke an associated loader in the binary representation.
11. A method of generating an optimized binary representation of an object tree, comprising:
 - receiving markup language information corresponding to an object tree;
 - generating an optimized binary representation of the markup language information; and
 - exposing the optimized binary representation to external resources.
12. A method according to claim 11, wherein the markup language information comprises extensible application markup language information.
13. A method according to claim 11, wherein the object tree corresponds to user interface elements.

14. A system according to claim 11, wherein the optimization comprises at least one of optimizing the binary representation at least by encoding dimension information in the binary representation, indexing the first instance of a novel object type in the optimized binary representation, and embedding an identifier to invoke an associated loader in the binary representation.

15. A method according to claim 11, wherein the step of exposing comprises exposing the optimized binary representation via a comprises a serial interface.

16. A method according to claim 11, wherein the external resources to which the binary representation is exposed comprise application programming interfaces.

17. An optimized binary representation of an object tree, the optimized binary representation being generating according to a method of:

receiving markup language information corresponding to an object tree;

generating an optimized binary representation of the markup language information; and

exposing the optimized binary representation to external resources.

18. An optimized binary representation according to claim 17, wherein the markup language information comprises extensible application markup language information.

19. An optimized binary representation according to claim 17, wherein the object tree corresponds to user interface elements.

20. An optimized binary representation according to claim 17, wherein the optimization comprises at least one of optimizing the binary representation at least by encoding dimension information in the binary representation, indexing the first instance of a novel object type in the optimized binary representation, and embedding an identifier to invoke an associated loader in the binary representation

21. An optimized binary representation according to claim 17, wherein the step of exposing comprises exposing the optimized binary representation via a serial interface.

22. An optimized binary representation method according to claim 17, wherein the external resources to which the binary representation is exposed comprise application programming interfaces.

23. A system for generating an optimized binary representation of an object tree, comprising:

input interface means to receive markup language information corresponding to an object tree;

mapping means, the mapping means communicating with the input interface means to receive the markup language information and generate an optimized binary representation of the markup language information; and

output interface means, communicating with the mapping means, the output interface means exposing the optimized binary representation to external resources.

24. A system according to claim 23, wherein the markup language information comprises extensible application markup language information.

25. A system according to claim 23, wherein the object tree corresponds to user interface elements.
26. A system according to claim 23, wherein the optimization comprises at least one of optimizing the binary representation at least by encoding dimension information in the binary representation, indexing the first instance of a novel object type in the optimized binary representation, and embedding an identifier to invoke an associated loader in the binary representation.
27. A system according to claim 23, wherein the output interface means comprises serial interface means.
28. A system according to claim 23, wherein the external resources to which the binary representation is exposed comprise application programming interfaces.
29. A computer readable medium, the computer readable medium being readable to execute a method of generating an optimized binary representation of an object tree, the method comprising:
- receiving markup language information corresponding to an object tree;
 - generating an optimized binary representation of the markup language information; and
 - exposing the optimized binary representation to external resources.
30. A computer readable medium according to claim 29, wherein the markup language information comprises extensible application markup language information.
31. A computer readable medium according to claim 29, wherein the object tree corresponds to user interface elements.